ВИЩИЙ НАВЧАЛЬНИЙ ЗАКЛАД УКООПСПІЛКИ
«ПОЛТАВСЬКИЙ УНІВЕРСИТЕТ ЕКОНОМІКИ І ТОРГІВЛІ»

НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ БІЗНЕСУ ТА СУЧАСНИХ ТЕХНОЛОГІЙ

ФОРМА НАВЧАННЯ ДЕННА
КАФЕДРА МАТЕМАТИЧНОГО МОДЕЛЮВАННЯ ТА СОЦІАЛЬНОЇ ІНФОРМАТИКИ

Допускається до захисту

Завідувач кафедри _____О.О. Ємець
(підпис)
«_____»_____2020 р.

ПОЯСНЮВАЛЬНА ЗАПИСКА
ДО БАКАЛАВРСЬКОЇ РОБОТИ

на тему

«ОПТИМІЗАЦІЯ ВИКОРИСТАННЯ ОБЛАДНАННЯ: ПРОГРАМНА РЕАЛІЗАЦІЯ ТРЕНАЖЕРА ДИСТАНЦІЙНОГО КУРСУ «ПРОЕКТНЕ НАВЧАННЯ З КУРСУ «МЕТОДИ ОПТИМІЗАЦІЇ ТА ДОСЛІДЖЕННЯ ОПЕРАЦІЙ»

зі спеціальності 122 «Комп'ютерні науки»

Виконавець роботи Веретельник Микита Русланович
_____ «___»_____2021р.
(підпис)
Науковий керівник д.ф-м.н., проф. Ємець О.О.
_____ «___»_____2021р.
(підпис)

ПОЛТАВА 2021р.

ЗМІСТ

ВСТУП

На сьогоднішній день розвиток сучасного суспільства пов’язаний з інформатизацію. Усі сфери діяльності людства тісно пов’язані с інформаційними системами, винятком не стала й освіта. Найбільш перспективним напрямком вдосконалення навчального процесу у вищих навчальних закладах є його комп’ютеризація. Електронне навчання має ряд переваг, а саме, можливість вивчення матеріалу у будь-який час, більше можливостей подання матеріалу студенту у більш зрозумілій формі, одночасне навчання великої кількості студентів, а також менше навантаження на викладачів тощо. Тому не дивно, що електронне навчання, на сьогоднішній день використовується у багатьох країнах світу.

Одним із видів електронного навчання є використання електронних тренажерів. Це сучасний інструмент, який допомагає зробити навчання більш цікавим, варіативним та підходить для вирішення складних задач. Тому вони вважаються одними з найшвидших способів набуття необхідних навичок на практиці.

Актуальність реалізації такого тренажеру обумовлена в зацікавленості навчальних закладів покращити та полегшити навчальний процес для студентів та викладачів.

Мета роботи – створити тренажер, який допоможе студентам зрозуміти як складати математичну модель і за допомогою MS Excel знаходити рішення.

Об’єкт роботи – розробка тренажеру з побудови математичної моделі.

Предмет роботи – оптимізація використання обладнання: алгоритмізація та програмування елементів тренажера з побудови математичної моделі.

Структура пояснювальної записки до проекту: титульний аркуш, завдання до бакалаврської роботи, зміст, вступ, теоретична частина, інформаційний огляд, практична частина, висновки, список використаної літератури, додатки. Теоретична частина містить умову задачі. В практичній

частині висвітлюється розв’язок задачі за допомогою математичного пакету Excel, а також реалізація тренажеру.

Методи. Для створення тренажу використовується середовище програмування Visual Studio 2019 Proffesional на мові програмування C#. Для побудови моделі – метод математичного моделювання

Обсяг пояснювальної записки: 49 стор., в т.ч. основна частина 38 стор., додатки - 11 стор., джерел – 3 назви.

1. ПОСТАНОВКА ЗАДАЧІ

1.1 Постановка задачі розробки тренажера

Головною задачею бакалаврської роботи є реалізація тренажеру з теми «Оптимізація використання обладнання: алгоритмізація та програмування елементів тренажера з побудови математичної моделі».

Для створення тренажеру використовується середовище програмування Visual Studio 2019 Proffesional з використанням мови програмування C++.

Для розробки тренажеру використовується задача про оптимізацію використання обладнання, в якій необхідно знайти максимальний прибуток.

Для початку необхідно побудувати математичну модель та розв'язати її в програмному пакеті MS Excel.

Для виготовлення трьох видів виробів *A, B, C* використовується токарне, фрезерне, зварювальне обладнання. Затрати часу на обробку одного виробу для кожного з типів обладнання вказані в таблиці. В ній вказано загальний фонд робочого часу кожного з типів обладнання, а також прибуток від реалізації одного виробу кожного виду.

Таблиця 1.1 – Умова задачі

Тип обладнання	Витрати часу (верстато-год) на обробку виробу виду			Загальий фонд (год.)
	A	***B***	***C***	
Фрезерне	2	1	3	120
Токарне	1	7	4	280
Зварювальне	7	6	3	240
Прибуток за виріб (грн.)	10	11	13	

Потрібно вирішити, скільки виробів і якого виду треба виготовити підприємству, щоб прибуток від їх реалізації був би максимальним.

Із умови задачі, можна поставити основні задачі:

— необхідно скласти алгоритм роботи тренажеру;

— розробити блок схему, яка висвітлює роботу тренажеру та всі можливі варіанти її роботи;

— реалізувати тренажер програмно;

— тестування тренажеру на наявність помилок та на відповідність до поставленої задачі.

Також, є ряд технічних вимог, яким повинен відповідати тренажер, а саме:

— тренажер повинен підтримувати актуальні версії Windows, а саме Windows 7, Windows 10;

— тренажер повинен мати легкий процес інсталювання на комп’ютер користувача;

— необхідно створити зручний та зрозумілий інтерфейс тренажеру, який допоможе користувачеві швидше адаптуватися до застосунку.

2. ІНФОРМАЦІЙНИЙ ОГЛЯД

2.1 Огляд робіт, де розглянуте аналогічне до теми роботи завдання

В процесі написання бакалаврської роботи був розглянутий тренажер на тему «Оптимізація виробництва столів: програмна реалізація тренажера(моделювання) дистанційного курсу «Проектне навчання з курсу «Методи оптимізації та дослідження операцій» » (Мороз А.В.)[1].

2.2 Позитивні аспекти оглянутих робіт

— Можливість переходу до всіх етапів тренажеру;

— присутня можливість використання тренажеру офлайн, тобто без доступу до мережі інтернет.

2.3 Вади оглянутих робіт

— Не дуже зрозумілий інтерфейс, текст складно читати;

— виникали помилки при використанні;

— деякі питання були незрозумілі, відсутні підказки.

2.4 Необхідність та актуальність теми

Розроблений тренажер допоможе студенту у будь-який час та більш зрозуміло й швидше набути певні навички для розв’язання типових задач на оптимізацію та їх побудову в Microsoft Excel.

3. ТЕОРЕТИЧНА ЧАСТИНА

3.1 Складання математичної моделі

Тренажер розробляється на основі конкретної задачі максимізації.

Для виготовлення трьох видів виробів *A*, *B*, *C* використовується токарне, фрезерне, зварювальне обладнання. Затрати часу на обробку одного виробу для кожного з типів обладнання вказані в таблиці. В ній вказано загальний фонд робочого часу кожного з типів обладнання, а також прибуток від реалізації одного виробу кожного виду.

Таблиця 3.1 – Умова задачі

Тип обладнаня	Витрати часу (верстато-год) на обробку виробу виду			Загальий фонд (год.)
	A	***B***	***C***	
Фрезерне	2	1	3	120
Токарне	1	7	4	280
Зварювальне	7	6	3	240
Прибуток за виріб (грн.)	10	11	13	

Потрібно вирішити, скільки виробів і якого виду треба виготовити підприємству, щоб прибуток від їх реалізації був би максимальним.

Для складання математичної моделі, введемо невідомі. Нехай:

— X_1 – виготовлено продукції виду А шт.;

— X_2 – виду В шт.;

— X_3 – виду С шт.;

Записуємо обмеження. Фрезерне обладнання має загальний фонд часу 120 годин(год.), тому перше обмеження матиме наступний вигляд:

$$2x1 + x2 + 3x3 \leq 120.$$

Токарне обладнання має загальний фонд часу 280 год., обмеження матиме вигляд:

$$x1 + 7x2 + 4x3 \leq 280.$$

Зварювальне обладнання має загальний фонд часу 240 год., обмеження матиме вигляд:

$$7x1 + 6x2 + 3x3 \leq 240.$$

Виходячи з цього побудуємо математичну модель.

Цю функцію необхідно максимізувати. Цільова функція матиме вигляд:

$$F(x) = 10x_1 + 11x_2 + 13x_3 \rightarrow \max$$

$$\begin{cases} 2x1 + x2 + 3x3 \leq 120\,; \\ x1 + 7x2 + 4x3 \leq 280\,; \\ 7x1 + 6x2 + 3x3 \leq 240; \\ x1, x2, x3 \geq 0. \end{cases}$$

3.2 Розв’язок задачі в «Розв’язувачі»

Для розв’язування моделі пакетом Microsoft Excel, з використанням його надбудови “Поиск решений” заповнюємо таблицю, що наведена на рисунку 3.1.

	A	B	C	D	E
1		A	B	C	
2	Фрезерне	2	1	3	120
3	Токарне	1	7	4	280
4	Зварювал	7	6	3	240
5	Прибуток	10	11	13	

Рисунок 3.1 – Вихідні дані задачі

Вносимо початкові дані для значень x_i функції *F(x)*, спочатку вони рівні 0 (рисунок 3.2).

	A	B	C	D	E	F
1		A	B	C		
2	Фрезерне	2	1	3	120	0
3	Токарне	1	7	4	280	0
4	Зварювал	7	6	3	240	0
5	Прибуток	10	11	13		
6		0	0	0		
7	F(x)=	0				

Рисунок 3.2 – Вихідні значення невідомих задачі

У комірку зі значенням цільової функції вносимо формулу, за якою обчислюється її значення рисунок 3.3.

Рисунок 3.3 – Обчислення цільової функції

Комірку F2(обмеження) заповнюємо формулою зображеною на рисунку 3.4.

Рисунок 3.4 – Обчислення обмежень

За аналогією заповнюємо комірки F3-F4.

Виконуємо такі дії у вікні надбудови (рисунок 3.5). У полі “Оптимизировать целевую функцию” обираємо ту комірку, у якій буде виводитися значення цільової функції. В полі “Изменяя ячейки” вводимо значення змінних, тобто комірки з *x* ($B$6:$D$6).

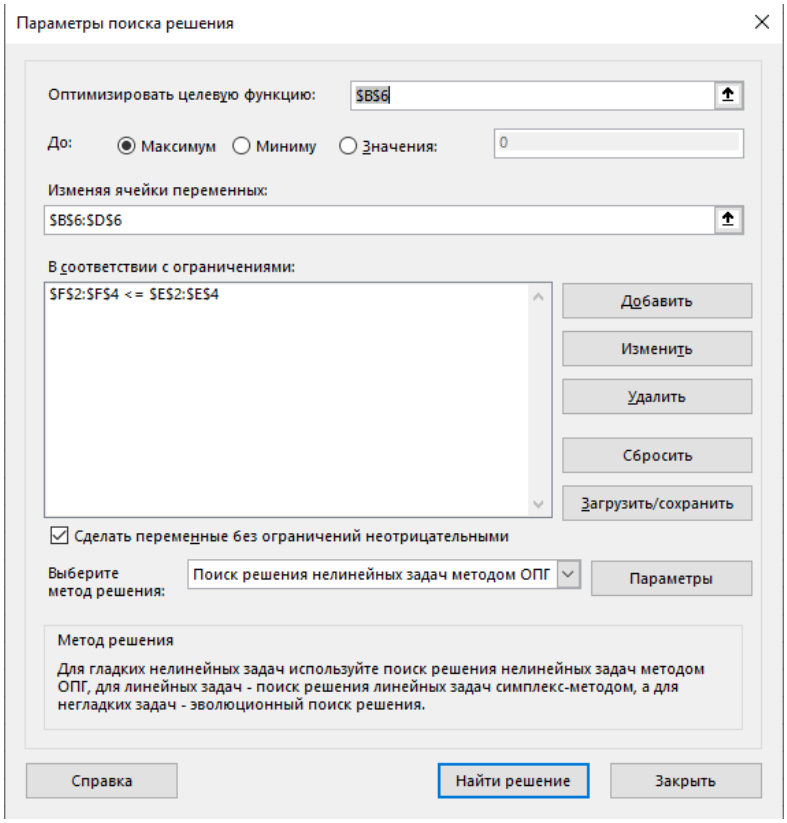


Рисунок 3.5 – Вікно надбудови

В полі “Ограничения” вносимо комірки з обмеженнями, потім натискаємо кнопку “Ok”(рисунок 3.6).

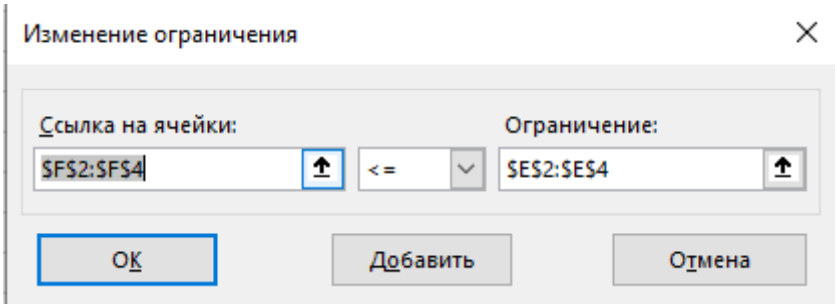


Рисунок 3.6 – Визначення обмежень

У вікні надбудови натискаємо “Найти решение”. Відкривається діалогове вікно, в якому натискаємо “Ok” (рисунок 3.7).

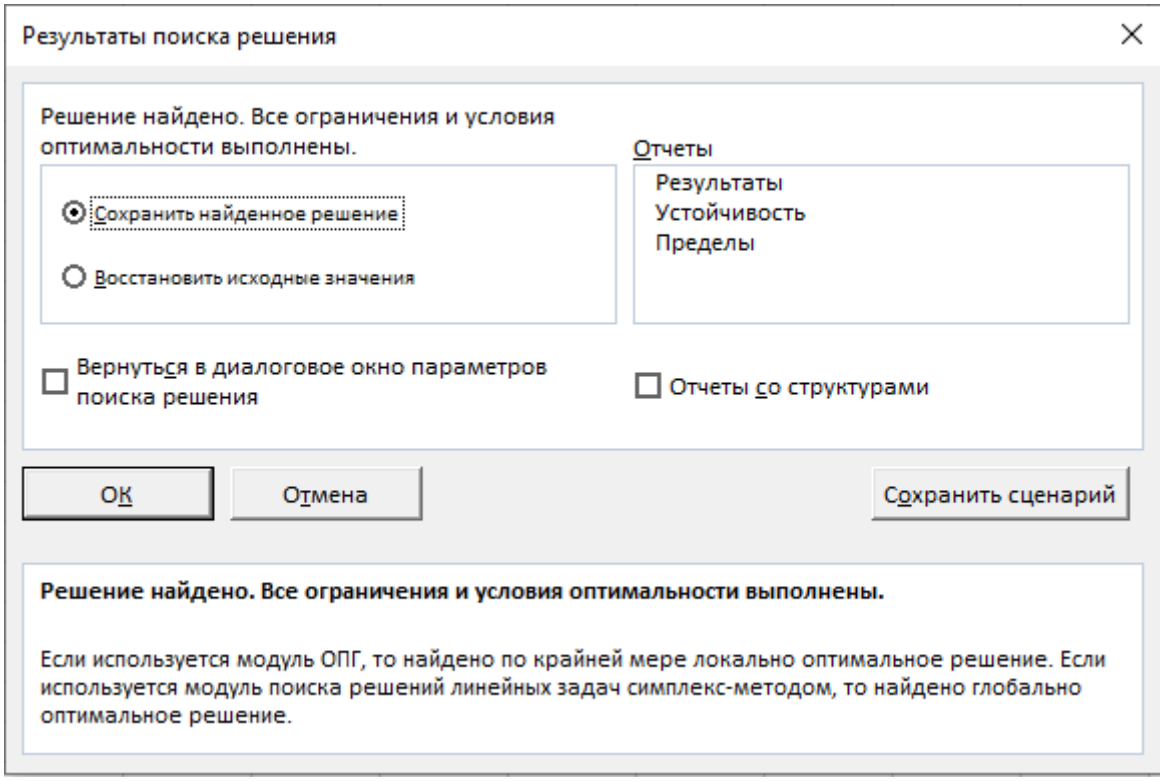


Рисунок 3.7 – Визначення параметрів звіту розв’язування задачі

У результаті обчислень отримуємо вихідні дані (рисунок 3.8).

	A	B	C	D	E	F
1		A	B	C		
2	Фрезерне	2	1	3	120	68,57143
3	Токарне	1	7	4	280	34,28571
4	Зварювал	7	6	3	240	240
5	Прибуток	10	11	13		
6		34,28571	0	0		
7	F(x)=	342,8571				

Рисунок 3.8 – Розв’язок задачі

3.3 Алгоритм роботи тренажера

Тренажер реалізований у форматі тесту. Присутні варіанти відповідей з однією або декількома вірними відповідями.

У разі неправильної відповіді тренажер виведе на екран повідомлення про помилку, а у раз правильного варіанту відповіді користувач перейде до наступного етапу.

Перейдемо до алгоритму тренажеру.

Крок 1. На екрані тренажеру виводиться текст, який ознайомлює користувача з задачею, дає інформацію про те, що потрібно зробити.

Далі виводиться запитання: “Що необхідно знайти в поставленій задачі?”

а) кількість виробів кожного типу;

б) кількість виробів якось певного типу;

в) кількість витраченого часу на виготовлення виробу;

г) кількість виробів 2-х різних типів.

Правильна відповідь: а.

Крок 2. “Що необхідно визначити в ході розв’язання задачі?”

а) оптимальний виріб одного типу, з точки зору мінімізації прибутку;

б) оптимальний виріб різних типів, з точки зору мінімізації прибутку;

в) кількість виробів кожного типу, що забезпечує максимізацію прибутку;

г) оптимальний виріб різних типів, з точки зору максимізації прибутку;

Правильна відповідь: в.

Крок 3. “Яке направлення цільової функції задачі?”

а) min max;

б) max;

в) max min;

г) min.

Правильна відповідь: б.

Крок 4. “Що необхідно максимізувати?”

а) сумарний прибуток, від реалізації всіх виробів;

б) час, відведений на виготовлення виробу;

в) прибуток, від реалізації виробу типу А;

г) час, відведений на виготовлення виробу типу А.

Правильна відповідь: а.

Крок 5. “Скільки видів виробів в даній задачі?”

а) 1;

б) 2;

в) 3;

г) 4.

Правильна відповідь: в.

Крок 6. “Як буде виглядати обмеження для фрезерного обладнання?”

а) $2x1 + x2 + 3x3 \leq 120.$;

б) $x1 + 7x2 + 4x3 \leq 280.$;

в) $7x1 + 6x2 + 3x3 \leq 240.$;

Правильна відповідь: а.

Крок 7. “Як буде виглядати обмеження для зварювального обладнання?”

а) $2x1 + x2 + 3x3 \leq 120.$;

б) $x1 + 7x2 + 4x3 \leq 280.$;

в) $7x1 + 6x2 + 3x3 \leq 240.$;

Правильна відповідь: в.

Крок 8. “Як буде виглядати обмеження для токарного обладнання?”

а) $2x1 + x2 + 3x3 \leq 120.$;

б) $x1 + 7x2 + 4x3 \leq 280.$;

в) $7x1 + 6x2 + 3x3 \leq 240.$;

Правильна відповідь: б.

Крок 9. “Ми маємо всі обмеження, тепер необхідно записати цільову функцію задачі. Як вона буде виглядати?”

Правильна відповідь: $F(x)= 10x_1 + 11x_2 + 13x_3 \rightarrow \max$.

Крок 10. На екран виводиться математична модель задачі.

Крок 11. Користувач переходить на другу частину тренажеру де дана задача розв’язується в Microsoft Excel.

Крок 12. Заповнюємо таблицю, що наведена на рисунку 3.9.

	A	B	C	D	E
1		A	B	C	
2	Фрезерне	2	1	3	120
3	Токарне	1	7	4	280
4	Зварювал	7	6	3	240
5	Прибуток	10	11	13	

Рисунок 3.9 – Вихідні дані задачі

Крок 13.Вносимо початкові дані для значень x_i функції $F(x)$, спочатку вони рівні 0 (рисунок 3.10).

	A	B	C	D	E	F
1		A	B	C		
2	Фрезерне	2	1	3	120	0
3	Токарне	1	7	4	280	0
4	Зварювал	7	6	3	240	0
5	Прибуток	10	11	13		
6		0	0	0		
7	F(x)=	0				

Рисунок 3.10 – Вихідні значення невідомих задачі

Крок 14. У комірку зі значенням цільової функції вносимо формулу, за якою обчислюється її значення рисунок 3.11.

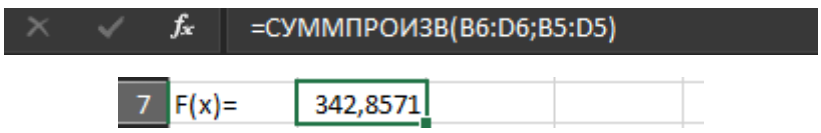


Рисунок 3.11 – Обчислення цільової функції

Крок 15. Комірку F2(обмеження) заповнюємо формулою зображеною на рисунку 3.12. За аналогією заповнюємо комірки F3-H4.

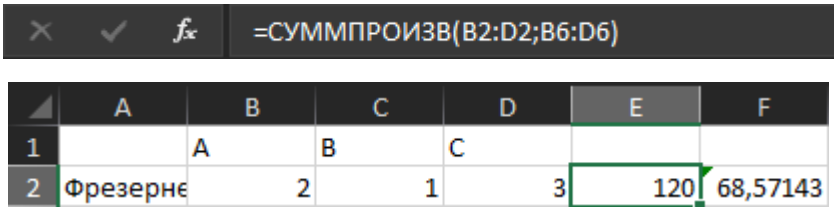


Рисунок 3.12 – Обчислення обмежень

Крок 16. У полі “Оптимизировать целевую функцию” обираємо ту комірку, у якій буде виводитися значення цільової функції. В полі “Изменяя ячейки” вводимо значення змінних, тобто комірки з *x* ($B$6:$D$6) (рисунок 3.13).

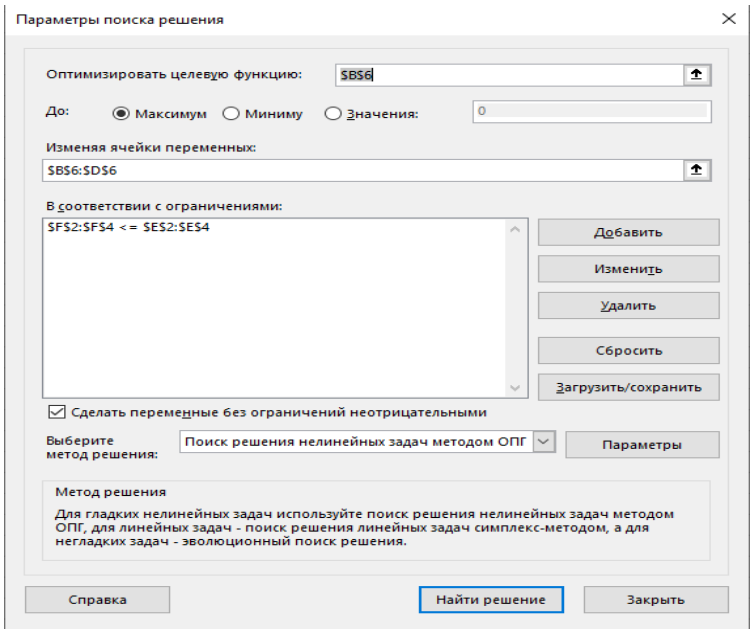


Рисунок 3.13 – Визначення параметрів надбудови

Крок 17. В полі “Ограничения” вносимо комірки з обмеженнями, потім натискаємо кнопку “Ok”(рисунок 3.14).

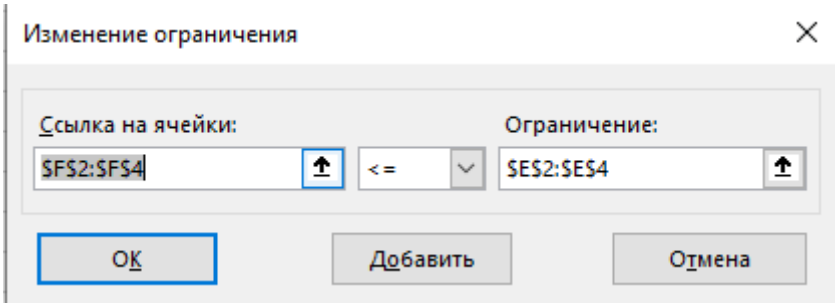


Рисунок 3.14 – Визначення обмежень

Крок 18. У вікні надбудови натискаємо “Найти решение”. Відкривається діалогове вікно, в якому натискаємо “Ok” (рисунок 3.15).

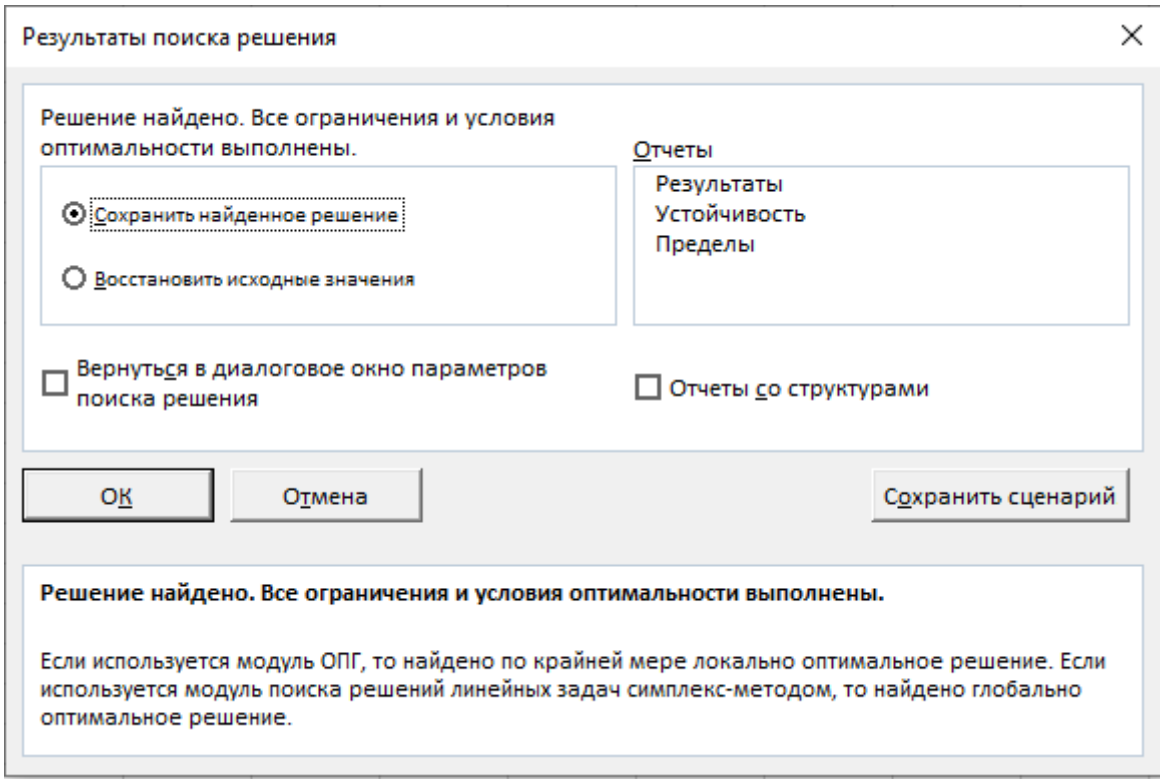


Рисунок 3.15 – Визначення параметрів звіту розв’язування задачі

Крок 19. У результаті обчислень отримуємо вихідні дані (рисунок 3.16).

	A	B	C	D	E	F
1		A	B	C		
2	Фрезерне	2	1	3	120	68,57143
3	Токарне	1	7	4	280	34,28571
4	Зварювал	7	6	3	240	240
5	Прибуток	10	11	13		
6		34,28571	0	0		
7	F(x)=	342,8571				

Рисунок 3.16 – Розв’язок задачі

3.4 Блок-схема тренажера

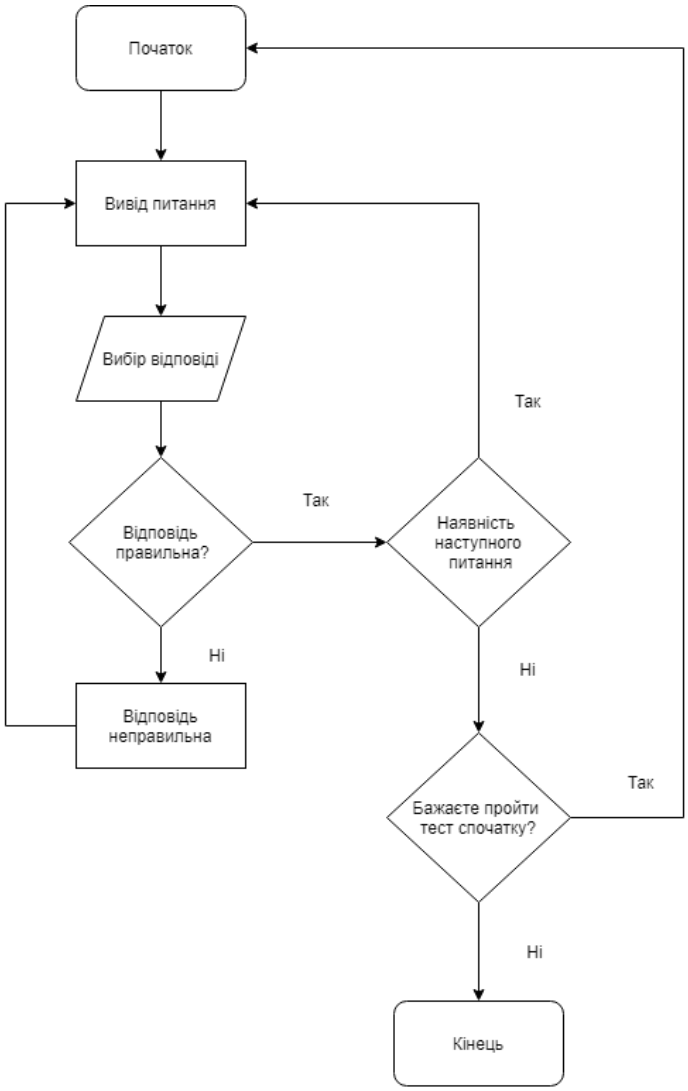


Рисунок 3.17 – Блок-схема тренажера

4. ПРАКТИЧНА ЧАСТИНА

4.1 Текст програми та її опис

Для написання цього тренажеру використовувалась мова програмування C#. Основним функціоналом тренажера є тестування, тому необхідно розробити алгоритм перевірки відповіді користувача та перехід до наступного або попереднього питання(Рисунок 4.1 і Рисунок 4.2). Весь код тренажеру знаходиться у додатку А.

```
Ссылок: 4
public partial class Tasks : MaterialSkin.Controls.MaterialForm
{
    String st;
    String st2;
    String st3;
    int correctAnswer;
    int questionNumber = 1;
    int totalQuestions;

    Ссылок: 2
    public Tasks()
    {
        InitializeComponent();
        comboBox1.KeyPress += (sender, e) => e.Handled = true;
        askQuestion(questionNumber);

        totalQuestions = 10;

    }

    Ссылок: 4
    private void CheckAnswerEvent(object sender, EventArgs e)
    {
        var senderObject = (Button)sender;

        int buttonTag = Convert.ToInt32(senderObject.Tag);

        if (buttonTag == correctAnswer)
        {
            questionNumber++;
            askQuestion(questionNumber);
        }
        else
        {
            MessageBox.Show("Відповідь не правильна!");
        }
    }
```

Рисунок 4.1 – Алгоритм перевірки відповіді

```
private void askQuestion(int qnum)
{
    switch (qnum)
    {
        case 1:
            lblQuestion.Text = "Що необхідно знайти в поставленій задачі?";

            button1.Text = "Кількість виробів кожного типу";
            button2.Text = "Кількість виробів будь-якого типу";
            button3.Text = "Кількість витраченого часу на виготовлення виробу";
            button4.Text = "Кількість виробів 2-х різних типів";

            correctAnswer = 2;

            break;

        case 2:
            lblQuestion.Text = "Що необхідно визначити в ході розв'язання задачі?";

            button1.Text = "оптимальний виріб одного типу, з точки зору мінімізації прибутку";
            button2.Text = "оптимальний виріб різних типів, з точки зору мінімізації прибутку";
            button3.Text = "оптимальний виріб одного типу, з точки зору максимізації прибутку";
            button4.Text = "оптимальний виріб різних типів, з точки зору максимізації прибутку";

            correctAnswer = 3;

            break;

        case 3:
            lblQuestion.Text = "Яке направлення цільової функції задачі?";

            button1.Text = "min max";
            button2.Text = "max";
            button3.Text = "max min";
            button4.Text = "min";

            correctAnswer = 2;

            break;
```

Рисунок 4.2 – Набір питань для тесту

Алгоритм виглядає наступним чином: ми беремо тег кнопки, іншими словами ідентифікатор кнопки й порівнюємо його з правильною відповіддю. Правильну відповідь ми визначаємо за допомогою змінної correctAnswer. Наприклад, якщо правильна відповідь рівна 2, то кнопка з тегом 2 буде правильною відповіддю. Якщо ми обираємо неправильну відповідь, то на екран виводиться повідомлення з помилкою й користувачу необхідно спробувати обрати іншу відповідь.

Також в тесті є питання, в якому необхідно вручну ввести відповідь (Рисунок 4.3).

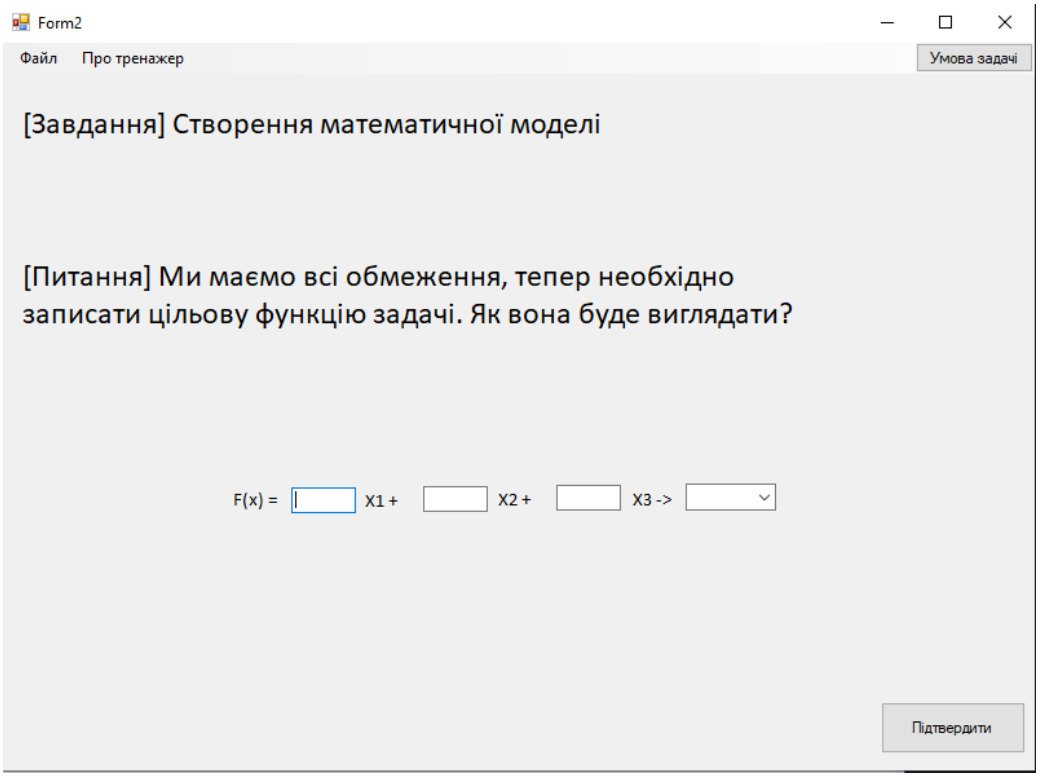


Рисунок 4.3 – Питання з вводом відповіді

У цьому питанні користувачу необхідно ввести дані для цільової функції. Оскільки нам потрібно, щоб користувач ввів тільки числа та обрав правильне направлення функції, необхідно зробити деякі перевірки (Рисунок 4.5).

```
private void button5_Click(object sender, EventArgs e)
{
    if (textBox1.Text == "" || textBox2.Text == "" || textBox3.Text == "" || comboBox1.SelectedIndex == -1)
    {
        MessageBox.Show("Перевірте правильність вводу");
    }
    else
    {
        st = textBox1.Text.ToString();
        st2 = textBox2.Text.ToString();
        st3 = textBox3.Text.ToString();

        int res1 = Convert.ToInt32(st);
        int res2 = Convert.ToInt32(st2);
        int res3 = Convert.ToInt32(st3);

        if (res1 == 10 && res2 == 11 && res3 == 13 && comboBox1.SelectedIndex == 1)
        {
            questionNumber++;
            askQuestion(questionNumber);
        }
        else
        {
            MessageBox.Show("Відповідь не правильна!");
        }

    }
}
```

Рисунок 4.5 – Перевірка вводу

По-перше, ми перевіряємо, чи ввів користувач щось у поле вводу. Якщо ні, користувач отримає повідомлення про помилку. Потім перевіряємо правильність вводу. Якщо все правильно, то користувач перейде до наступного питання, а якщо ні, отримає повідомлення про помилку. На рисунку 4.6 можна побачити обмеження для вводу даних. Користувач може вводи тільки цифри та використовувати кнопку backspace.

```
ссылка: 1
private void textBox1_KeyPress(object sender, KeyPressEventArgs e)
{
    char number = e.KeyChar;
    if (!Char.IsDigit(number) && number != 8)
    {
        e.Handled = true;
    }

}

ссылка: 1
private void textBox2_KeyPress(object sender, KeyPressEventArgs e)
{
    char number = e.KeyChar;
    if (!Char.IsDigit(number) && number != 8)
    {
        e.Handled = true;
    }
}

ссылка: 1
private void textBox3_KeyPress(object sender, KeyPressEventArgs e)
{
    char number = e.KeyChar;
    if (!Char.IsDigit(number) && number != 8)
    {
        e.Handled = true;
    }
}
```

Рисунок 4.6 – Обмеження вводу

За таким алгоритмом працює увесь тренажер.

4.2 Реалізація роботи тренажера

Початковий екран тренажеру на рисунку 4.7.

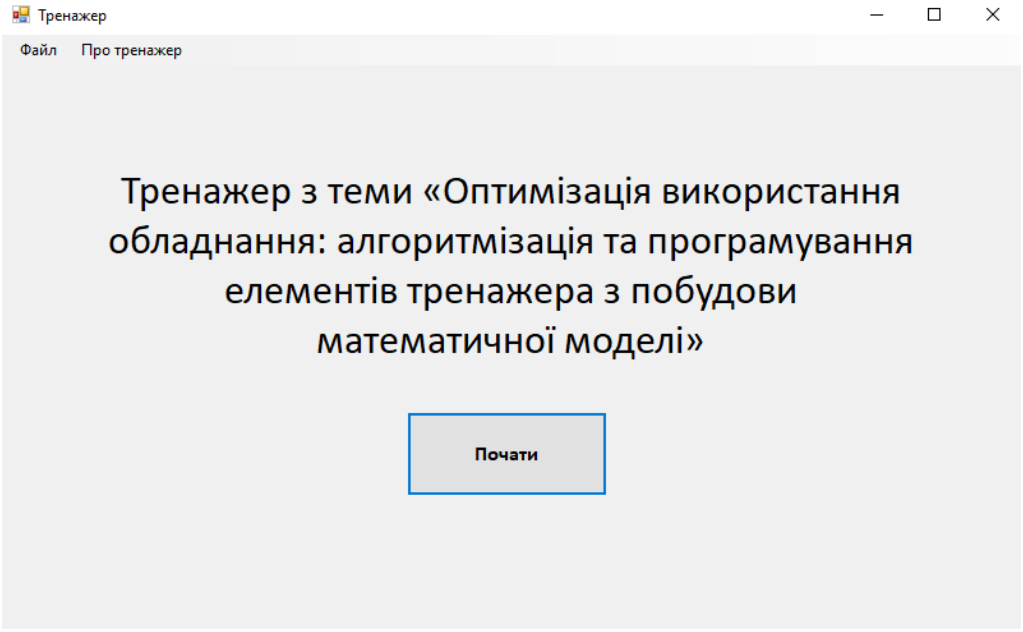
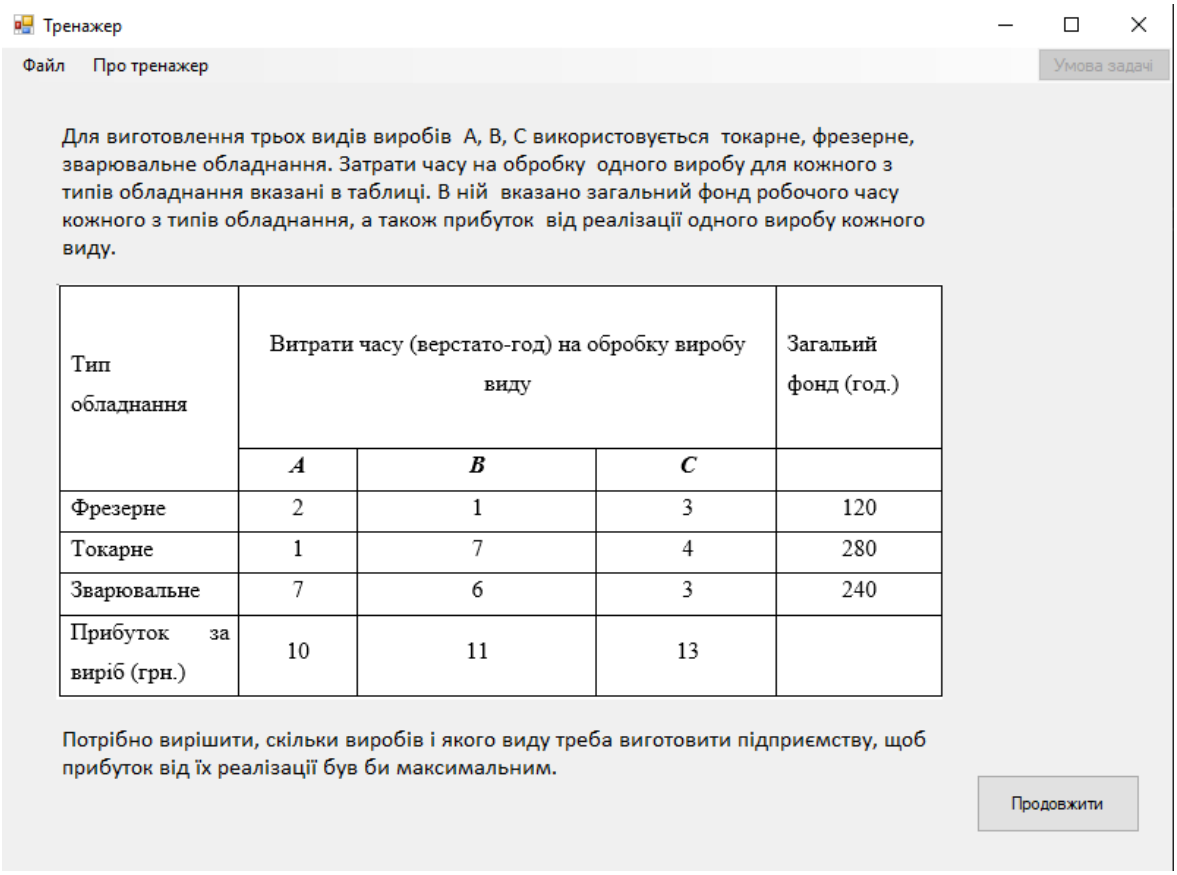


Рисунок 4.7 – Початковий екран

Екран с умовою задачі на рисунку 4.8.



Тип обладнання	Витрати часу (верстато-год) на обробку виробу виду			Загальний фонд (год.)
	A	*B*	*C*	
Фрезерне	2	1	3	120
Токарне	1	7	4	280
Зварювальне	7	6	3	240
Прибуток за виріб (грн.)	10	11	13	

Рисунок 4.8 – Умова задачі

Екран з неправильною відповіддю на рисунку 4.9.

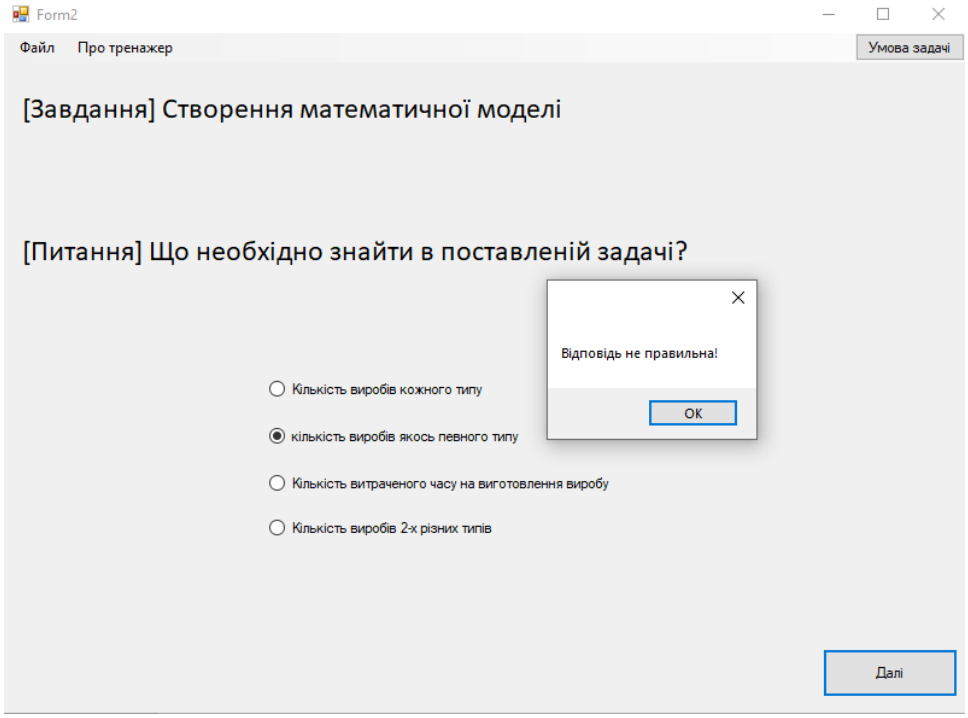


Рисунок 4.9 – Неправильна відповідь

Екран з питанням, де необхідно вручну ввести правильну відповідь. На рисунку 4.10 зображена помилка, коли поля для вводу порожні.

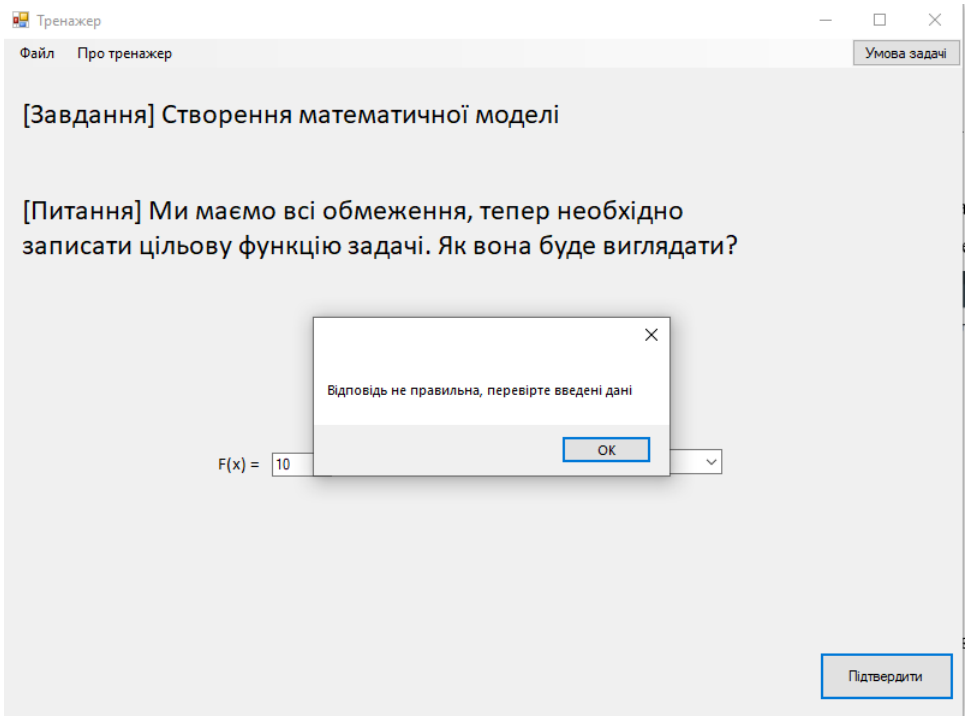


Рисунок 4.10 – Помилка при вводі даних

На рисунку 4.11 зображений екран з розв’язком задачі в Microsoftt Excel.

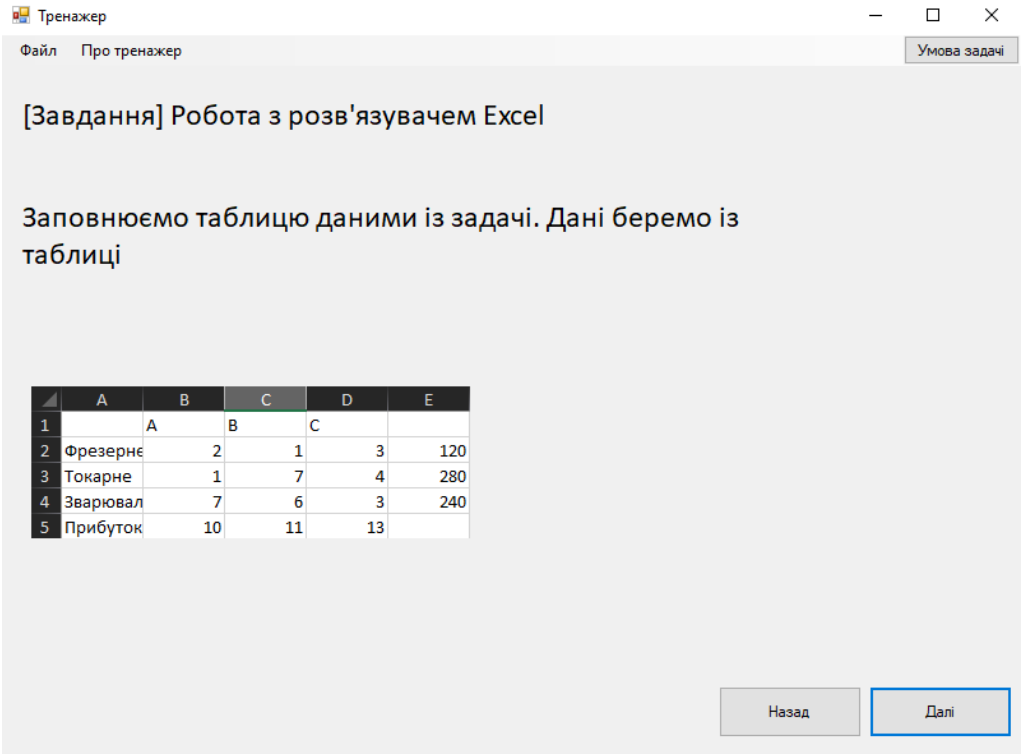


Рисунок 4.11 – Розв`язок у Microsoft Excel

На рисунку 4.12 зображену завершення роботи тренажера. Користувачу пропонується завершити роботу з тренажером або почати спочатку.

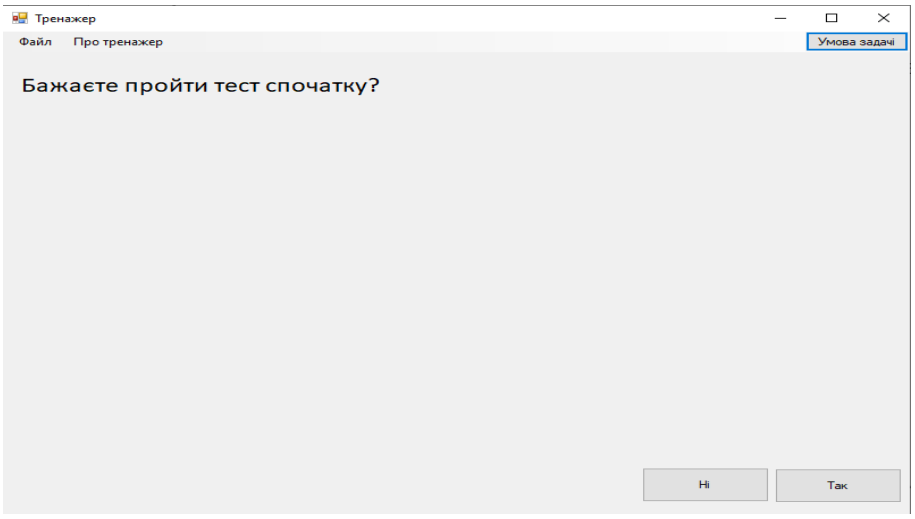
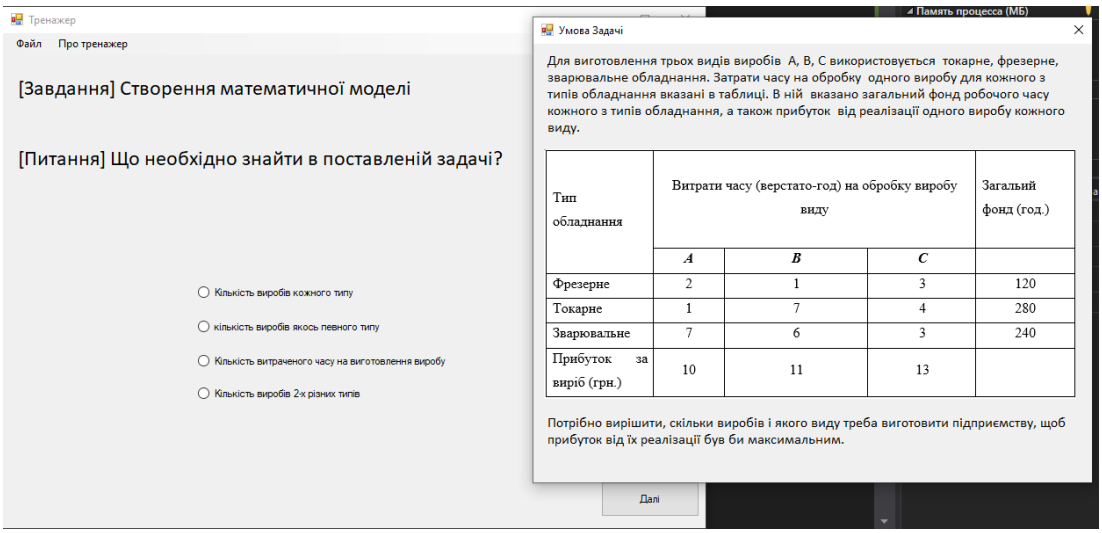


Рисунок 4.12 – Кінець тесту

У користувача на всіх етапах є можливість побачити умову задачі. Це необхідно для комфортнішого проходження тестування, оскільки на деякі питання неможливо дати відповідь без умови задачі перед очима. Для того, щоб

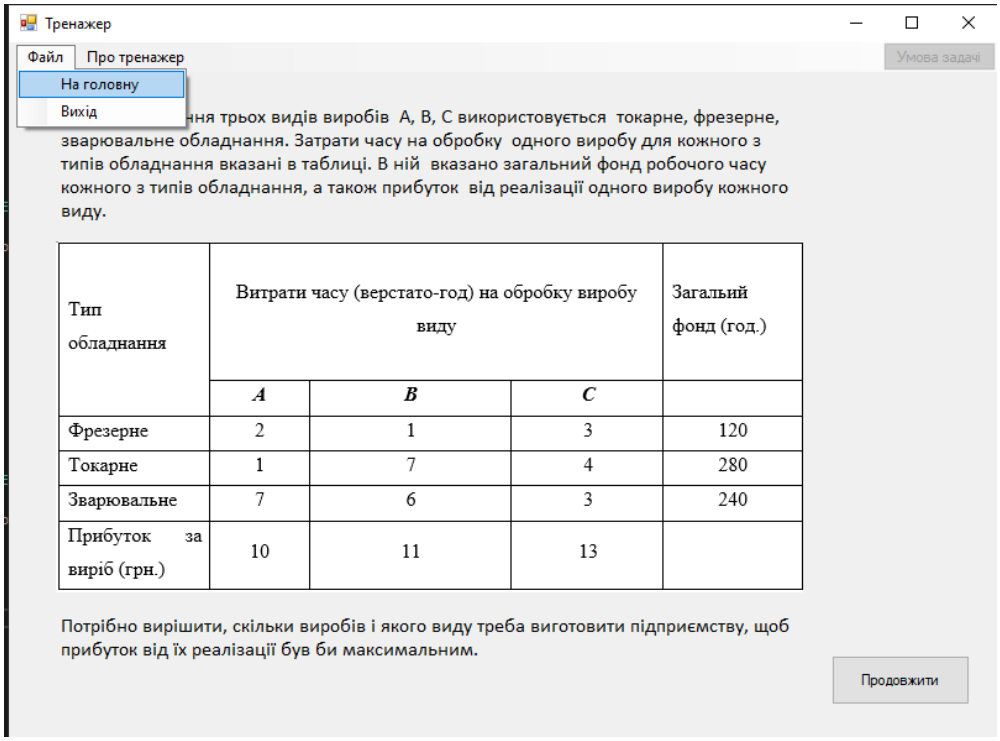
побачити умову задачі, необхідно натиснути на кнопку «Умова задачі», яка знаходиться у правому верхньому кутку вікна. Після натискання кнопки відкривається вікно з умовою задачі. Це вікно буде відкрите до моменту, поки користувач сам не закриє це вікно або не закінчить тестування (див. рис.4.13).



Тип обладнання	Витрати часу (верстато-год) на обробку виробу виду			Загальний фонд (год.)
	A	*B*	*C*	
Фрезерне	2	1	3	120
Токарне	1	7	4	280
Зварювальне	7	6	3	240
Прибуток за виріб (грн.)	10	11	13	

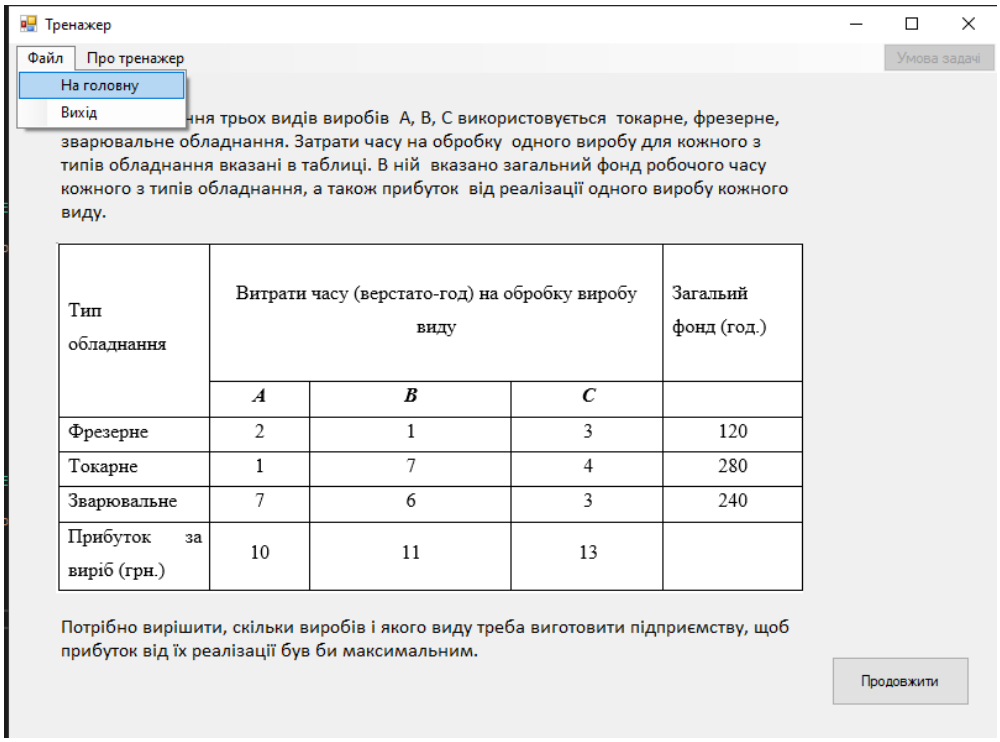
Рисунок 4.13 – Вікно «Умова задачі»

Також у користувача є можливість в будь-який момент повернутися на головний екран тренажера. Для цього необхідно натиснути у меню на вкладку «Файл» й обрати пункт «На головну». Вкладка зображена на рисунку 4.14. Користувач отримає попередження перед тим, як перейти на головний екран, це необхідно для того, якщо користувач випадково натиснув на кнопку або в якийсь момент передумав повертатися на головний екран. Діалогове вікно, яке з’являється після натискання кнопки «На головну» знаходиться на рисунку 4.15.



Тип обладнання	Витрати часу (верстато-год) на обробку виробу виду			Загальний фонд (год.)
	A	*B*	*C*	
Фрезерне	2	1	3	120
Токарне	1	7	4	280
Зварювальне	7	6	3	240
Прибуток за виріб (грн.)	10	11	13	

Рисунок 4.14 – Вкладка «На головну»



Тип обладнання	Витрати часу (верстато-год) на обробку виробу виду			Загальний фонд (год.)
	A	*B*	*C*	
Фрезерне	2	1	3	120
Токарне	1	7	4	280
Зварювальне	7	6	3	240
Прибуток за виріб (грн.)	10	11	13	

Рисунок 4.15 – Діалогове вікно після натискання кнопки «На головну»

В меню також присутня кнопка «Про тренажер». При натисканні на цю кнопку користувач побачить діалогове вікно, в якому буде висвітлена

інформація про розробника тренажера, наукового керівника та тему з назвою університету. Діалогове вікно можна побачити на рисунку 4.16.

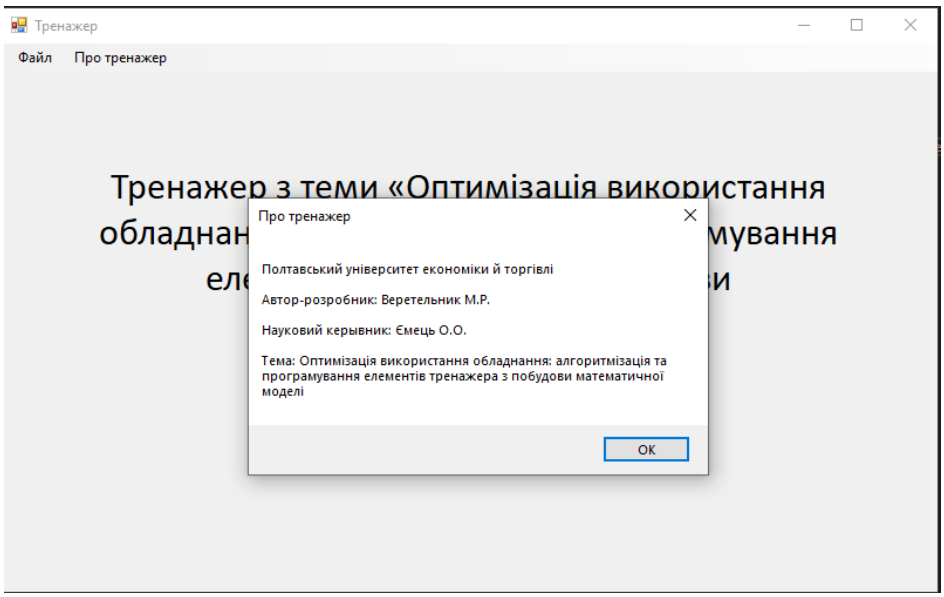


Рисунок 4.16 – Діалогове вікно «Про тренажер»

4.3 Тестування. Дослідження можливостей програмної реалізації

При запуску тренажера ми бачимо вікно (див. рисунок 4.7), яке дає змогу користувачу зрозуміти, для чого призначений програмний застосунок. Тренажер запускається без помилок. При натисканні кнопки «Почати» користувач переходить до умови задачі (див. рисунок 4.8).

На етапі тестування у користувача відсутня можливість повертатися до попереднього питання. Для переходу до наступного питання необхідно обрати правильно відповідь, тільки тоді можливо перейти до наступного питання. В іншому випадку на екрані буде з’являтися повідомлення про помилку.

На питанні, де користувачу необхідно вводити відповідь з клавіатури, тренажер повинен швидко й правильно реагувати на дії користувача. Тренажер повинен давати змогу вводити тільки цифри, при цьому введені числа мають бути правильними (рисунки 4.17, 4.18).

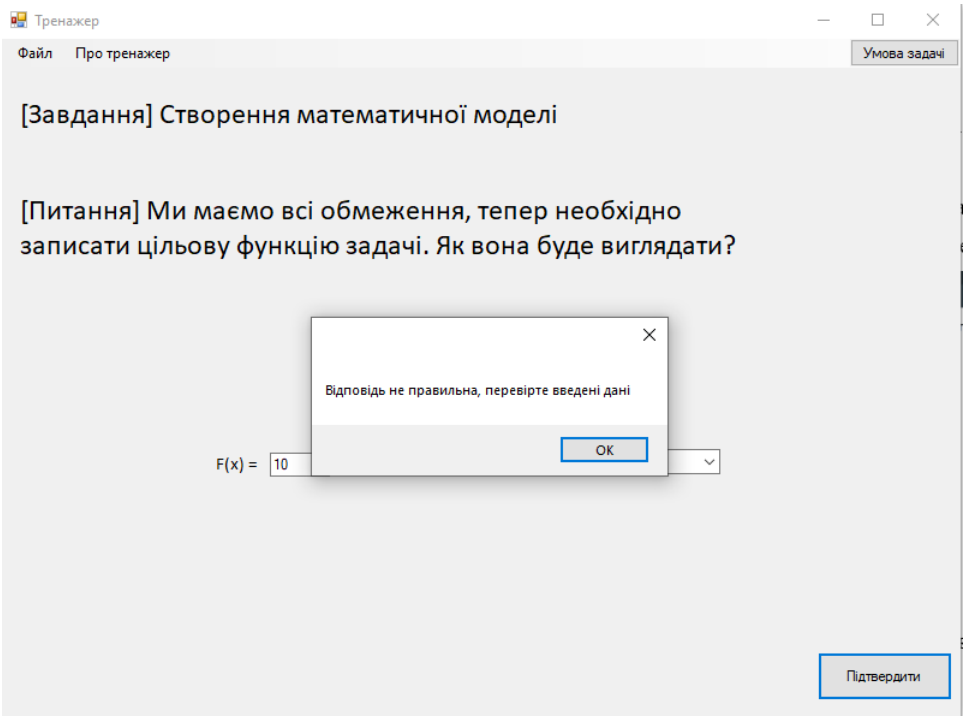


Рисунок 4.17 – Помилка, коли поля пусті

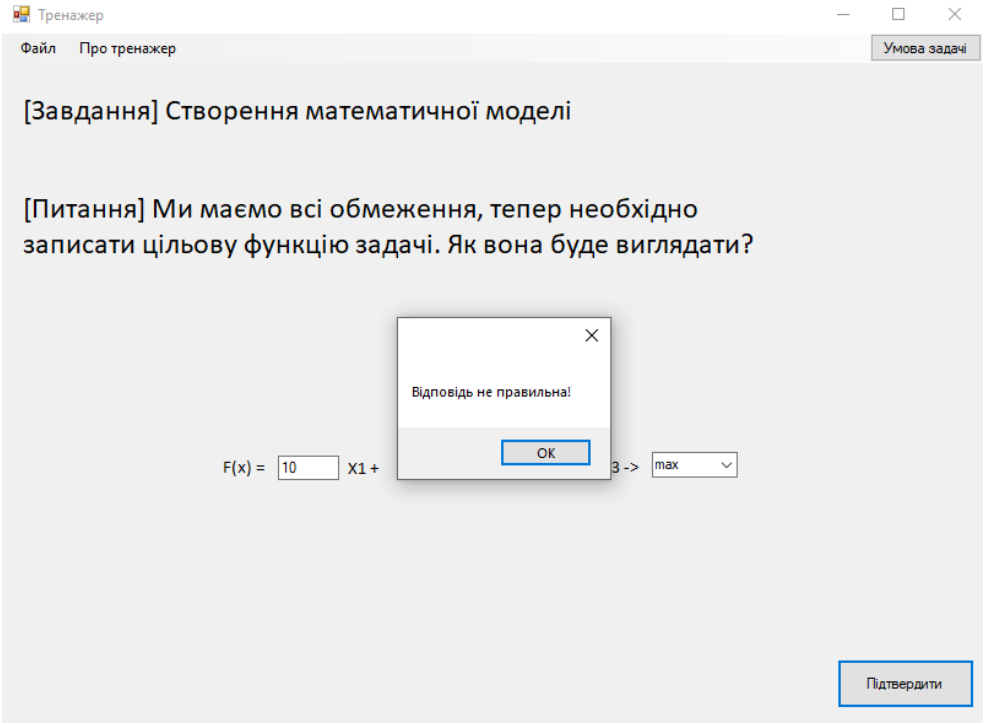


Рисунок 4.18 – Помилка, коли відповідь неправильна

Коли користувач переходить до частини, де показується розв’язок задачі в Microsoft Excel, тренажер повинен надати змогу повернутися до попереднього кроку (рисунок 4.19). На першому кроці кнопка «Назад» відсутня(рисунок 4.20).

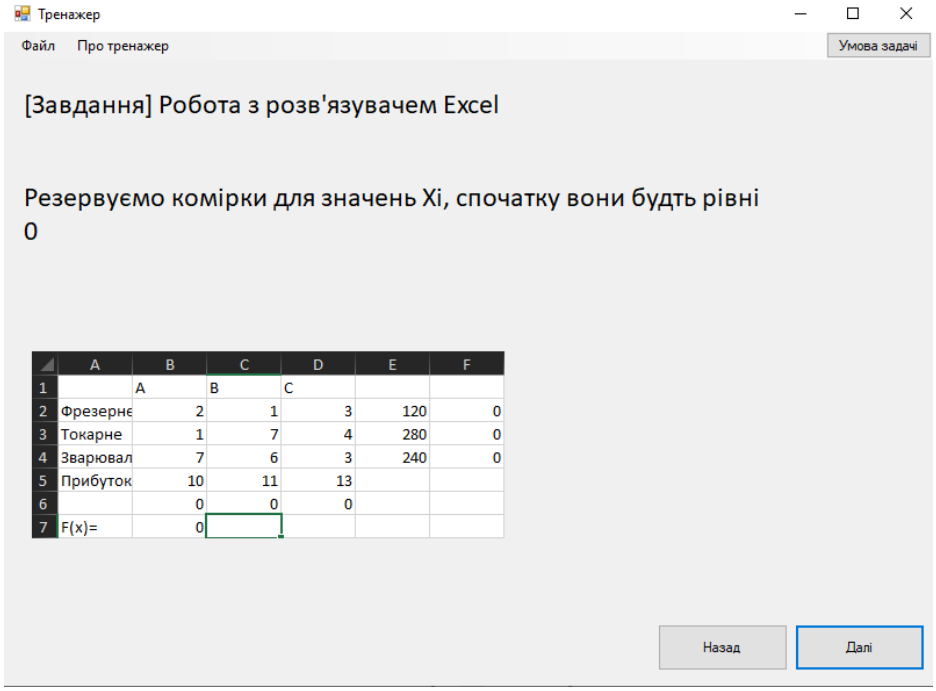


Рисунок 4.19 – Кнопка «Назад»

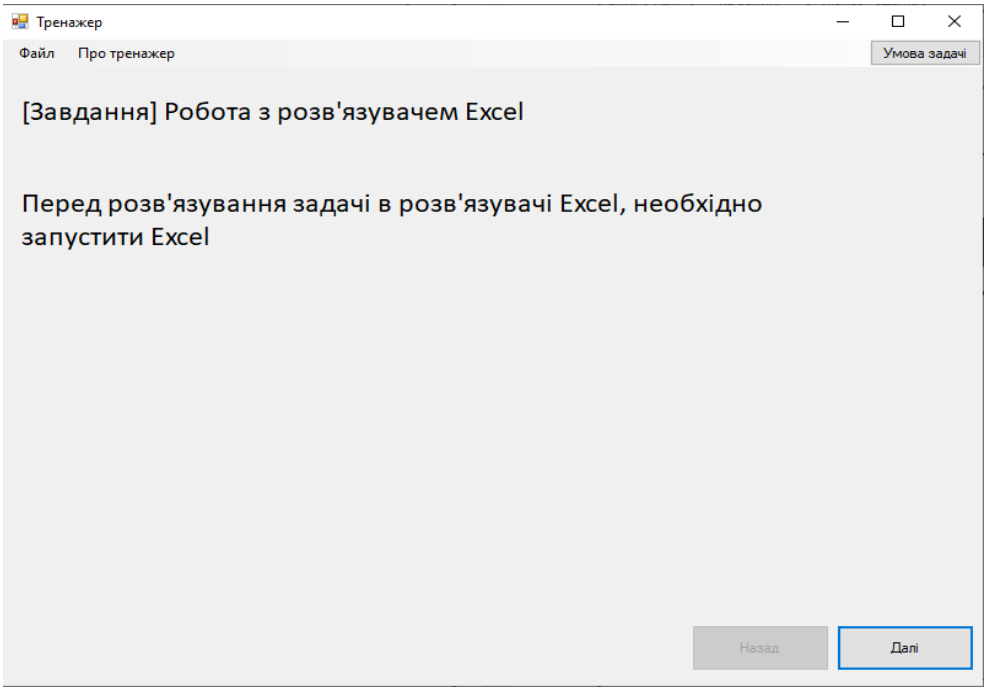
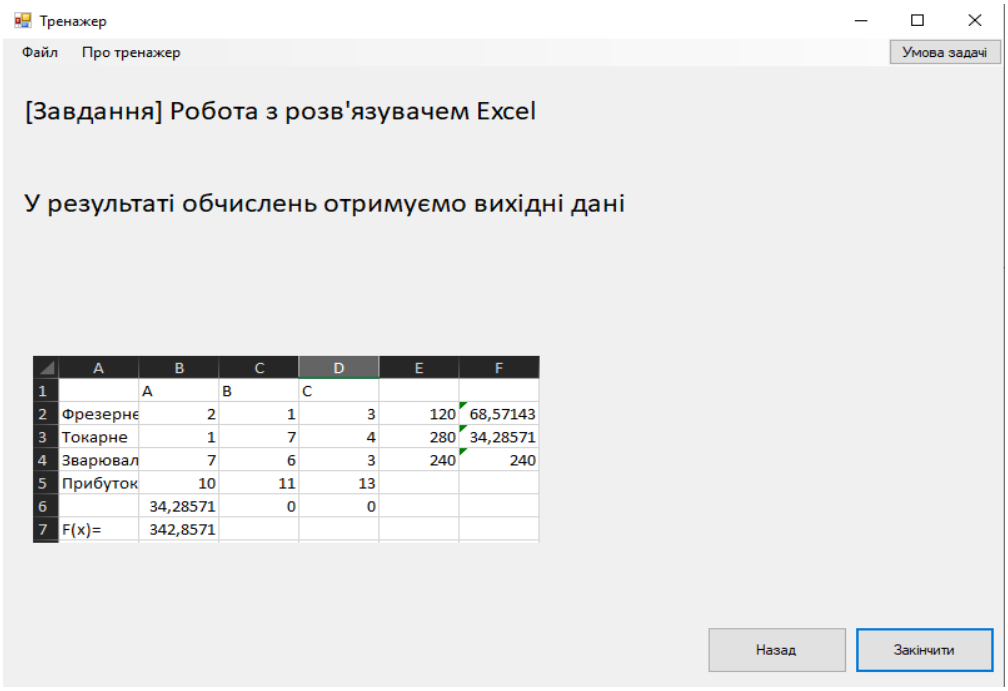


Рисунок 4.20 – Початковий екран розв’язку задачі в Microsoft Excel

На останньому кроці кнопка «Далі» повинна змінюватися на «Закінчити» (рисунок 4.21).



	A	B	C	D	E	F
1		A	B	C		
2	Фрезерне	2	1	3	120	68,57143
3	Токарне	1	7	4	280	34,28571
4	Зварювал	7	6	3	240	240
5	Прибуток	10	11	13		
6		34,28571	0	0		
7	F(x)=	342,8571				

Рисунок 4.21 – Останній етап

При завершенні роботи користувачу пропонується закінчити роботу або почати спочатку (рисунок 4.22).

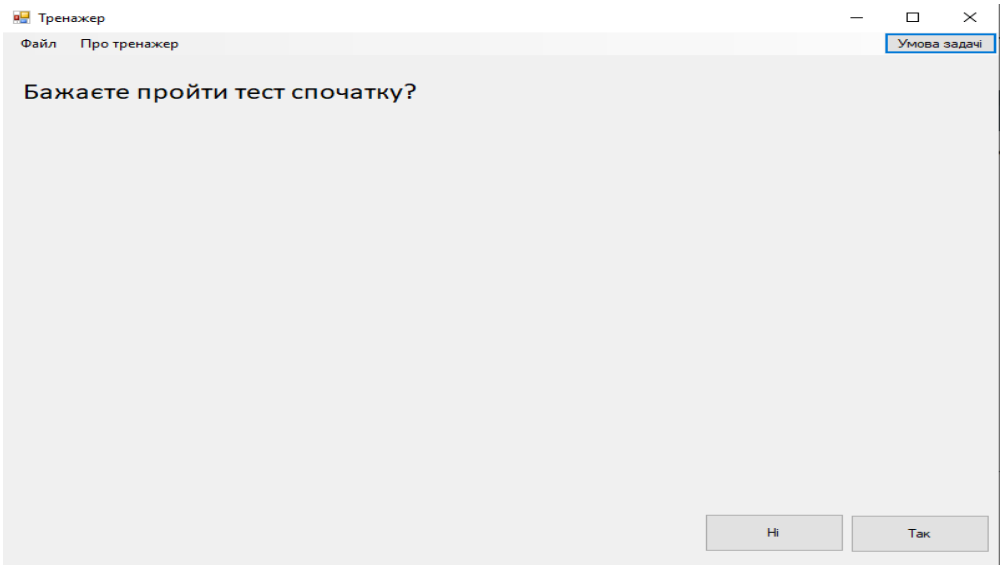


Рисунок 4.22 – Завершення роботи тренажеру

При натисканні кнопки «Умова задачі», користувач повинен побачити вікно з умовою задачі. Необхідно було провести декілька тестувань. Користувач повинен мати можливість бачити цю кнопку лише при проходженні тестування, тому на головному екрані її бути не повинно. Як

можна побачити на рисунку 4.7, ця кнопка відсутня. Після натискання цієї кнопки повинно з’явитися вікно, а кнопка «Умова задачі» повинна стати неактивною. Це необхідно для того, щоб користувач не міг визвати безліч вікон з умовою задачі. Як можна побачити на рисунку 4.13, кнопка неактивна й вікно відображене на моніторі користувача. Вікно «Умова задачі» повинно закриватися, коли користувач закінчує проходження тестування. При тестуванні це вікно залишалося активним після закінчення тестування, що є некритичним недоліком. Цей недолік необхідно виправити в наступних оновленнях. Демонстрація недоліку можна побачити на рисунку 4.23.

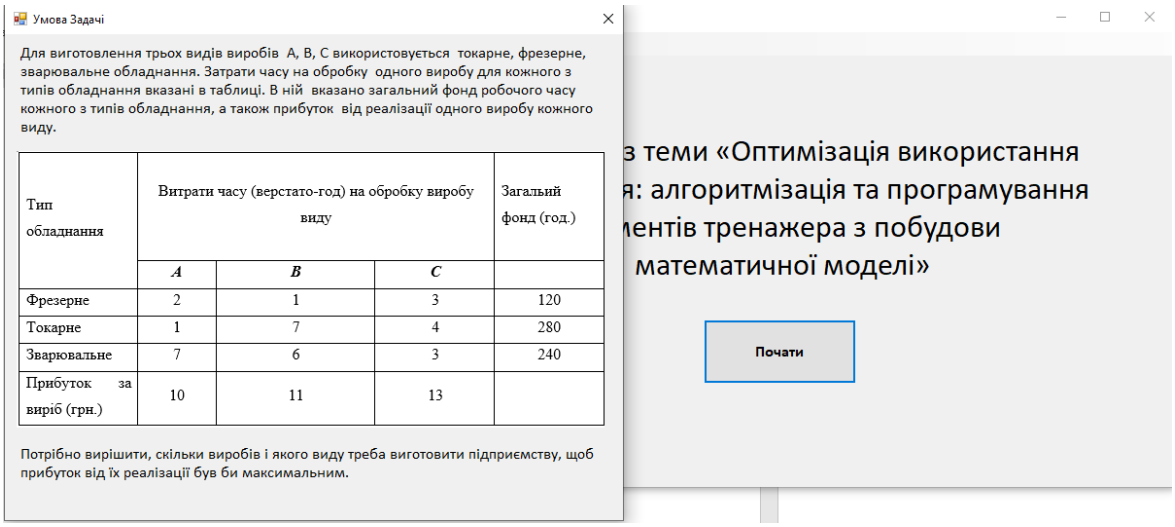


Рисунок 4.23 – Недолік вікна «Умова задачі»

Якщо користувач хоче закінчити роботу з тренажером або просто повернутися на головний екран, то він повинен побачити діалогове вікно з попередженням з двома варіантами відповіді: «Так», «Ні». У разі, якщо користувач обирає варіант «Так», то тренажер повинен закінчити роботу, а у випадку, якщо користувач хоче повернутися на головний екран, повернутися на головний екран. У разі, якщо користувач обирає варіант «Ні», то тренажер продовжує роботу у тому місці в якому закінчив, тобто тренажер не виконує ніяких дій. Діалогове вікно повинне закритися. При тестуванні виникло декілька помилок. В деяких ситуаціях діалогове вікно не відображалося зовсім,

у деяких відображалося декілька разів. Також були випадки, коли діалогове вікно з’являлося там, де це не потрібно. Це достатньо критичні помилки й вони повинні бути виправлені в найближчих оновленнях. На рисунку 2.24 можна побачити діалогове вікно при завершенні роботи з тренажером.

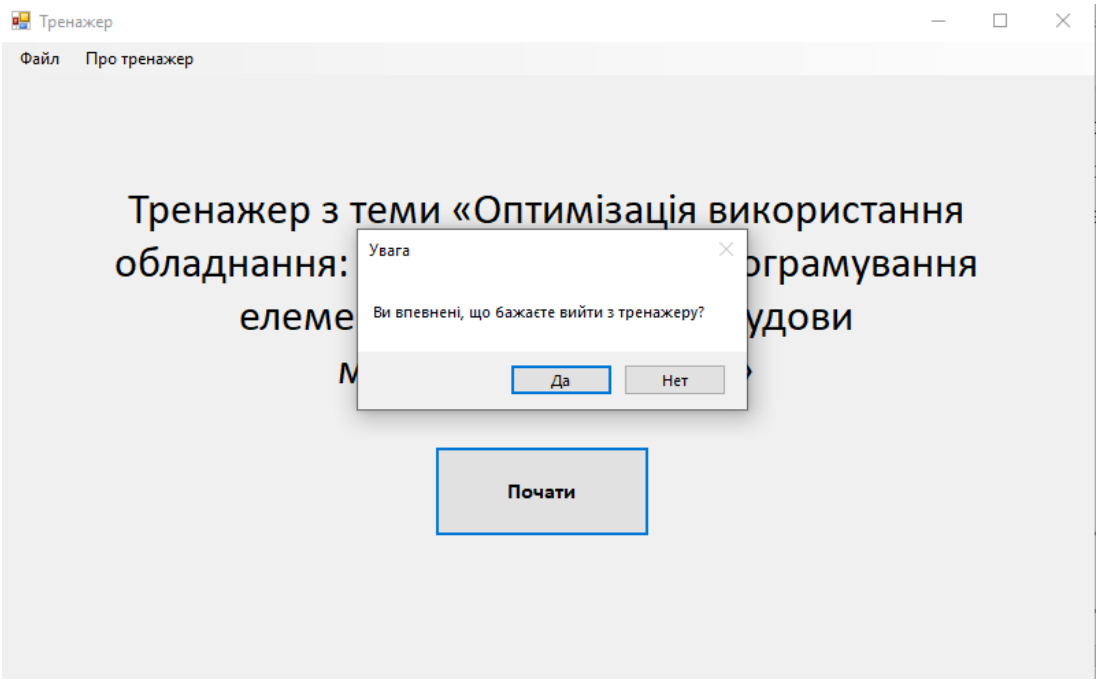


Рисунок 4.24 – Діалогове вікно при завершенні роботи з тренажером

При тестуванні також виникали випадки, коли при завершенні роботи тренажера, він не закривався повністю. Тренажер залишався запущеним, що є критичним недоліком, так як він може навантажувати систему й користувач просто не буде знати, що тренажер не закритий. Цей недолік необхідно виправити в найближчих оновленнях.

ВИСНОВКИ

Тренажер створений з використанням мови програмування C# та середовища розробки Visual Studio 2019. Він допомагає надати та покращити навички студента в створенні математичної моделі в задачах оптимізації.

Під час виконання бакалаврської роботи був створений тренажер з теми «Оптимізація використання обладнання » для дистанційного курсу «Проектне навчання з дисципліни «Методи оптимізації та дослідження операцій».

Слід зауважити, що тренажер розроблений з ціллю продемонструвати можливості й алгоритм тренажера. Тренажер рекомендується ретельно перевірити до використання на практиці, оскільки він може містити помилки й недоліки.

У майбутньому планується вдосконалення тренажеру, а саме додавання наступного функціоналу:

— панель адміністратора для викладача, з можливістю створення нових тестів;
— покращення та вдосконалення інтерфейсу;
— можливість вибору тесту;
— використання бази даних.

Основна мета та завдання бакалаврської роботи виконано, тренажер створено й він має базову функціональність, яка допомагає складати математичну модель задачі й знаходить рішення за допомогою прикладного пакету Microsoft Excel.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Мороз А. Пояснювальна записка до бакалаврської роботи на тему оптимізація виробництва столів: Програмна реалізація тренажера (моделювання) дистанційного курсу «Проектне навчання з курсу «Методи оптимізації та дослідження операцій» [Електронний ресурс] / Артур Мороз – Режим доступу до ресурсу: http://dspace.puet.edu.ua/handle/123456789/9017.
2. Борута І. Пояснювальна записка до бакалаврської роботи на тему тренажер з теми «Відношення. область визначення, область значень, граф, матриця відповідності, переріз за елементами» Дистанційного навчального курсу «Дискретна математика» та розробка його програмного забезпечення [Електронний ресурс] / Іван Борута – Режим доступу до ресурсу: http://dspace.puet.edu.ua/handle/123456789/10352.
3. Величко А. Пояснювальна записка до бакалаврської роботи на тему програмне забезпечення для тренажера з теми «Способи задання мов» дистанційного навчального курсу «Теорія програмування» [Електронний ресурс] / Артур Величко – Режим доступу до ресурсу: http://dspace.puet.edu.ua/handle/123456789/9031.

ДОДАТОК А. Код тренажеру

```
        int correctAnswer;
        int questionNumber = 1;
        int totalQuestions;
        int radio = 0;
        public Label lbl1;
        String st;
        String st2;
        String st3;

        public Form2()
        {
            InitializeComponent();
            askQuestion(questionNumber);
            comboBox1.KeyPress += (sender, e) => e.Handled = true;
            totalQuestions = 20;
            this.panel2.AutoScroll = true;
        }

        private void askQuestion(int qnum)
        {
            switch (qnum)
            {
                case 1:
                    lbl_Task.Text = "[Завдання] Створення математичної моделі";
                    lbl_question.Text = "[Питання]" + " " + "Що необхідно знайти в
поставленій задачі?";

                    radioButton1.Text = "Кількість виробів кожного типу";
                    radioButton2.Text = "кількість виробів якось певного типу";
                    radioButton3.Text = "Кількість витраченого часу на виготовлення виробу";
                    radioButton4.Text = "Кількість виробів 2-х різних типів";
                    correctAnswer = 1;
                    panel2.Visible = false;
                    btn_Back.Visible = false;
```

```
                    break;


                case 2:
                    lbl_question.Text = "[Питання]" + " " + "Що необхідно визначити в ході
розв’язання задачі?";


                    radioButton1.Text = "оптимальний виріб одного типу, з точки зору
мінімізації прибутку";
                    radioButton2.Text = "оптимальний виріб різних типів, з точки зору
мінімізації прибутку";
                    radioButton3.Text = "кількість виробів кожного типу, що забезпечує
максимізацію прибутку";
                    radioButton4.Text = "оптимальний виріб різних типів, з точки зору
максимізації прибутку";
                    radioButton1.Focus();
                    correctAnswer = 3;


                    break;


                case 3:
                    lbl_question.Text = "[Питання]" + " " + "Яке направлення цільової
функції задачі?";


                    radioButton1.Text = "min max";
                    radioButton2.Text = "max";
                    radioButton3.Text = "max min";
                    radioButton4.Text = "min";
                    radioButton1.Focus();
                    correctAnswer = 2;


                    break;


                case 4:
                    lbl_question.Text = "[Питання]" + " " + "Що необхідно максимізувати?";


                    radioButton1.Text = "максимізувати сумарний прибуток";
                    radioButton2.Text = "час, відведений на виготовлення виробу";
                    radioButton3.Text = "прибуток, від реалізації виробу типу А";
```

```
                    radioButton4.Text = "час, відведений на виготовлення виробу типу А";
                    radioButton1.Focus();
                    correctAnswer = 1;

                    break;

                case 5:
                    lbl_question.Text = "[Питання]" + " " + "Скільки видів виробів в даній
задачі?";

                    radioButton1.Text = "1";
                    radioButton2.Text = "2";
                    radioButton3.Text = "3";
                    radioButton4.Text = "4";
                    radioButton1.Focus();
                    correctAnswer = 3;

                    break;

                case 6:
                    lbl_question.Text = "[Питання]" + " " + "Як буде виглядати обмеження для
фрезерного обладнання?";

                    radioButton1.Text = "2x1+x2+3x3≤120";
                    radioButton2.Text = "x1+7x2+4x3≤280";
                    radioButton3.Text = "7x1+6x2+3x3≤240";
                    radioButton4.Visible = false;
                    radioButton1.Focus();
                    correctAnswer = 1;

                    break;

                case 7:
                    lbl_question.Text = "[Питання]" + " " + "Як буде виглядати обмеження для
зварювального обладнання?";

                    radioButton1.Text = "2x1+x2+3x3≤120";
```

```
                    radioButton2.Text = "x1+7x2+4x3≤280";
                    radioButton3.Text = "7x1+6x2+3x3≤240";
                    radioButton4.Visible = false;
                    radioButton1.Focus();
                    correctAnswer = 3;

                    break;

                case 8:
                    lbl_question.Text = "[Питання]" + " " + "Як буде виглядати обмеження для токарного обладнання?";

                    radioButton1.Text = "2x1+x2+3x3≤120";
                    radioButton2.Text = "x1+7x2+4x3≤280";
                    radioButton3.Text = "7x1+6x2+3x3≤240";
                    radioButton4.Visible = false;
                    radioButton1.Focus();
                    correctAnswer = 2;

                    break;

                case 9:
                    lbl_question.Text = "[Питання]" + " " + "Ми маємо всі обмеження, тепер необхідно записати цільову функцію задачі. Як вона буде виглядати?";
                    radioButton1.Visible = false;
                    radioButton2.Visible = false;
                    radioButton3.Visible = false;
                    radioButton4.Visible = false;
                    panel1.Visible = true;
                    btn_Next.Visible = false;
                    break;

                case 10:
                    lbl_question.Text = "[Питання]" + " " + "Математична модель задачі створено. Тепер перейдемо до розв'язування задачі в розв'язувачі Microsoft Excel";
                    radioButton1.Visible = false;
                    radioButton2.Visible = false;
```

```
                radioButton3.Visible = false;
                radioButton4.Visible = false;
                panel1.Visible = false;
                pictureBox2.Visible = true;
                btn_Next.Visible = true;
                btn_Confirm.Visible = false;
                btn_Next.Text = "Перейти до розв'язувача";
                break;



            case 11:
                lbl_Task.Text = "[Завдання] Робота з розв'язувачем Excel";
                lbl_question.Text = "Перед розв'язування задачі в розв'язувачі Excel,
необхідно запустити Excel";
                panel2.Visible = true;
                pictureBox1.Visible = false;
                pictureBox2.Visible = false;
                btn_Back.Visible = true;
                btn_Back.Enabled = false;
                btn_Next.Text = "Далі";
                btn_Back.Text = "Назад";
                break;


            case 12:
                lbl_question.Text = "Заповнюємо таблицю даними із задачі. Дані беремо із
таблиці";
                pictureBox1.Image = Properties.Resources.one;
                pictureBox1.Visible = true;
                btn_Back.Visible = true;
                btn_Back.Enabled = true;
                btn_Next.Text = "Далі";
                btn_Back.Text = "Назад";
                this.panel2.AutoScroll = false;
                break;


            case 13:
```

```
                    lbl_question.Text = "Резервуємо комірки для значень Xi, спочатку вони будть рівні 0";

                    pictureBox1.Image = Properties.Resources.two;

                    break;


                case 14:

                    lbl_question.Text = "У комірку зі значенням цільової функції вносимо формулу, за якою обчислюється її значення, на малюнку в комірках вже є значення, але на данному етапі вони будуть рівні 0";

                    pictureBox1.Image = Properties.Resources.three;


                    break;


                case 15:

                    lbl_question.Text = "Комірку F2(обмеження) заповнюємо формулою зображеною на рисунку. За аналогією заповнюємо комірки F3-H4. Після введення формул значення в комірках будуть рівні 0";

                    pictureBox1.Image = Properties.Resources.four;


                    break;


                case 16:

                    lbl_question.Text = "У полі “Оптимизировать целевую функцию” обираємо ту комірку, у якій буде виводитися значення цільової функції. В полі “Изменяя ячейки” вводимо значення змінних, тобто комірки з x ($B$6:$D$6) ";

                    pictureBox1.Image = Properties.Resources.fifth;

                    this.panel2.AutoScroll = true;

                    break;


                case 17:

                    lbl_question.Text = "В полі “Ограничения” вносимо комірки з обмеженнями, потім натискаємо кнопку “Ok”";

                    pictureBox1.Image = Properties.Resources.six;

                    this.panel2.AutoScroll = false;

                    break;


                case 18:

                    lbl_question.Text = "У вікні надбудови натискаємо “Найти решение”. Відкривається діалогове вікно, в якому натискаємо “Ok” ";
```

```
            pictureBox1.Image = Properties.Resources.seven;
            this.panel2.AutoScroll = true;
            break;

        case 19:
            lbl_question.Text = "У результаті обчислень отримуємо вихідні дані";
            pictureBox1.Image = Properties.Resources.eight;
            this.panel2.AutoScroll = false;
            btn_Next.Text = "Закінчити";
            break;

        case 20:
            lbl_Task.Text = "Бажаєте пройти тест спочатку?";
            lbl_question.Visible = false;
            pictureBox1.Visible = false;
            btn_Next.Visible = false;
            btn_Back.Visible = false;
            button2.Visible = true;
            button3.Visible = true;

            break;
    }
}

private void Form2_Load(object sender, EventArgs e)
{
    panel1.Visible = false;
    pictureBox2.Visible = false;
    btn_TaskInfo.Enabled = false;
    panel3.BringToFront();
    button2.Visible = false;
    button3.Visible = false;
}

private void btn_Next_Click(object sender, EventArgs e)
{
```

```
        if (radioButton1.Checked == true)
        {
            radio = 1;
        }
        else if (radioButton2.Checked == true)
        {
            radio = 2;
        }
        else if (radioButton3.Checked == true)
        {
            radio = 3;
        }
        else if (radioButton4.Checked == true)
        {
            radio = 4;
        }

        if (radio == correctAnswer)
        {
            questionNumber++;
            askQuestion(questionNumber);
        }
        else
        {
            MessageBox.Show("Відповідь не правильна!");
        }

        if(btn_Next.Text == "Перейти до розв'язувача")
        {
            Form3 excel = new Form3();
            excel.Show();
            this.Close();
        }

    }
```

```
        private void btn_TaskInfo_Click(object sender, EventArgs e)
        {
            foreach (Form TaskInfo in Application.OpenForms)
            {
                if (TaskInfo.Name == "TaskInfo")
                {
                    return;
                }

            }

            TaskInfo task = new TaskInfo();
            task.Show();
            task.btn1 = this.btn_TaskInfo;
            btn_TaskInfo.Enabled = false;
        }

        private void btn_Confirm_Click(object sender, EventArgs e)
        {
            if (textBox1.Text == "" || textBox2.Text == "" || textBox3.Text == "" ||
comboBox1.SelectedIndex == -1)
            {
                MessageBox.Show("Відповідь не правильна, перевірте введені дані");
            }
            else
            {
                st = textBox1.Text.ToString();
                st2 = textBox2.Text.ToString();
                st3 = textBox3.Text.ToString();

                int res1 = Convert.ToInt32(st);
                int res2 = Convert.ToInt32(st2);
                int res3 = Convert.ToInt32(st3);

                if (res1 == 10 && res2 == 11 && res3 == 13 && comboBox1.SelectedIndex == 1)
                {
```

```
            questionNumber++;
            askQuestion(questionNumber);
        }
        else
        {
            MessageBox.Show("Відповідь не правильна!");
        }

    }
}

private void button1_Click(object sender, EventArgs e)
{
    panel3.Visible = false;
    btn_TaskInfo.Enabled = true;
}

private void btn_Back_Click_1(object sender, EventArgs e)
{
    questionNumber--;
    askQuestion(questionNumber);
}

private void button2_Click(object sender, EventArgs e)
{
    Form2 main = new Form2();
    main.Show();
    this.Close();
}

private void button3_Click(object sender, EventArgs e)
{
    foreach (Form TaskInfo in Application.OpenForms)
    {
        if (TaskInfo.Name == "TaskInfo")
        {
```

```
                return;
            }
            else
            {
                this.Close();
            }

        }
        Form1 frm1 = new Form1();
        frm1.Show();
        this.Dispose();
    }

    private void наГоловнуToolStripMenuItem_Click(object sender, EventArgs e)
    {
        DialogResult result = MessageBox.Show("Ви впевнені, що бажаєте повернутися на головну?","Увага", MessageBoxButtons.YesNo);


        if (result == DialogResult.Yes)
        {
            Form1 frm1 = new Form1();
            frm1.Show();
            this.Dispose();
        }

        this.TopMost = true;
    }

    private void вихідToolStripMenuItem_Click(object sender, EventArgs e)
    {
        Application.Exit();
    }

}
```